UNISYS

A Series
SNA ENVOY
MT EMULATOR

Implementation and Operations Reference Manual

Release 13.04

Priced Item

September 1992

Printed in U S America
3790 3127-000

UNISYS

Product Information Announcement

• New release ○ Revision ○ Update ○ New Mail Code

Title

A Series SNA ENVOY MT Emulator Implementation and Operations Reference Manual (3790 3127-000)

This Product Information Announcement announces the release and availability of the *A Series SNA ENVOY MT Emulator Implementation and Operations Reference Manual* for release level 13.04, dated September 1992.

The MT Emulator is a Unisys software product that enables an IBM 3270 display station to function as a Unisys terminal in an SNA network.

This manual describes the following:

- installation and configuration
- the IBM 3270 display station and its data stream format
- MT Emulator functionality and software interfaces
- operation of the 3270 display station under emulation

To order copies

- United States customers, call Unisys Direct at 1-800-448-1424.
- International customers, contact your Unisys Sales Office.
- Unisys personnel, use the Electronic Literature Ordering (ELO) system.

Announcement Only:
MBZ

Announcement and attachments:

System: A Series
SNA ENVOY

Release: 13.04 September 1992

Part number: 3790 3127-000

DOCUMENTATION SURVEY

hank you for taking the time to help us evaluate the product information that we prepare for you.

)ocumentation Evaluation

lease indicate your degree of satisfaction by checking ıe appropriate box:	Very Unsatisfactory	Unsatisfactory	Neither Satisfactory Nor Unsatisfactory	Satisfactory	Very Satisfactory	Comments
. Technical accuracy						
. Completeness of coverage						
. Number of examples						
. Usefulness of examples						
. Usefulness of system message descriptions						
. Ease with which information can be located						
. Organization of document						
. Easy to understand						
. Usefulness of index						
0. Number of illustrations						
1. Usefulness of illustrations						
2. Well written						
3. Appearance of document						
4. Overall document						

. Which of the following best describes your use of the document?

_____ I read the entire document.
_____ I read only the sections applicable to what I was doing.
_____ I skimmed through the document.
_____ I looked up specific items only when I needed the information.

. How often do you use the document?

_____ Several times a day.
_____ Several times a week.
_____ Several times a month.
_____ Almost never.

. Which parts of the document do you use the most?

_____

_____

. What do you like about this document?

_____

_____

. What one thing could we do that would most improve the document?

_____

_____

(over)

Site Information

1. Name: _____

2. Company/location: _____

3. Title/Function: _____

4. Can we call and discuss the documents with you? ☐ Yes ☐ No Phone No. ( ) _____

5. Briefly describe your network configuration (if known):

_____

_____

_____

_____

Fold along dotted lines so that only mailing label shows and tape closed.

No Postage necessary if mailed in the United States

BUSINESS REPLY MAIL

First Class Mail Permit No.26 Southeastern PA 19398

Postage Will Be Paid By Addressee

UNISYS CORPORATION
ATTN: SPG PRODUCT INFORMATION
EAST COAST DEVELOPMENT CENTER ROOM K154
PO BOX 203
PAOLI PA 19301-9865

UNISYS

A Series
SNA ENVOY
MT EMULATOR

Implementation and Operations Reference Manual

Copyright © 1992 Unisys Corporation
All rights reserved.
Unisys is a registered trademark of Unisys Corporation.

Release 13.04

September 1992

Printed in U S America

Priced Item

3790 3127-000

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

RESTRICTED RIGHTS LEGEND. Use, reproduction, or disclosure is subject to the restrictions set forth in DFARS 252.227-7013 and FAR 52.227-14 for commercial computer software.

Correspondence regarding this publication should be forwarded to Unisys Corporation either by using the Business Reply Mail Form at the back of this manual or by addressing remarks directly to Unisys Corporation, East Coast Development Center, SPG Product Information, P.O. Box 203, Paoli PA 19301-9898, U.S.A.

SNA ENVOY, SNA EXPRESS/NJE, SNA PRINT SUPPORT, and MDM are trademarks of Unisys Corporation.

IBM is a registered trademark of International Business Machines Corporation.

Page Status

Page	Issue
iii	-000
iv	Blank
v through vi	-000
1 through 3	-000
4	Blank
5 through 15	-000
16	Blank
17 through 25	-000
26	Blank
27 through 35	-000
36	Blank
37 through 58	-000
A-1 through A-4	-000
B-1 through B-3	-000
B-4	Blank
Index-1 through Index-3	-000
Index-4	Blank

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.

Contents

Chapter 1

Introduction 1
Structure of this manual 2
Product Description 3

Chapter 2

Installation 5
MT EMULATOR Internal Files 6
Display Store File 6
Terminal Configuration File 6
Configuring The MT EMULATOR 7
Home Menu 7
Operational Environment Screen 8
EBCDIC Code Point Translation Chart 9
Terminal Options 11
Resource Requirements 15

Chapter 3

IBM 3270 Display Station 17
IBM 3270 Video 18
IBM 3270 Keyboard Functions 21

Chapter 4

IBM 3270 Data Stream 27
Commands 28
Attention Identifier (AID) 31
Orders 32

Chapter 5

MT EMULATOR Functional Description 37
Keyboard and Data Comm Cursor 37
Display Buffer 37
Terminal State 38
Video Character Set 38
Application Status Line 39
System Status Line 39
Page Formatting 40
Terminal Operations Controlled via Data Comm 42
Text Editing 42
Terminal Control Functions 44

Search Mode 44
Video Control 44
Peripheral Control Commands 44
Key Programming 45
Terminal Configuration 45
NDL II Editing of Data Stream 46

Chapter 6

IBM 3270 in Emulation Mode 47
Invoking Unisys Terminal Functions 47
Default PF and PA Key Settings 50
Real-Time Configuration Mode 51
Customizing the PF Keys 53

Chapter 7

MT EMULATOR Software Interface 55
OPEN MTEMULATOR SESSION 55
CLOSE MTEMULATOR SESSION 56
MTEMULATOR 56
Request Word Layout 57
Result Word Layout 58

Appendix A

Unisys Terminal Escape Sequences A-1

Appendix B

Unisys Terminal Configuration B-1

Chapter 1
Introduction

This document describes the various aspects of emulating a Unisys terminal device on an IBM 3270 information display system using the MT EMULATOR. Throughout this manual, any reference to a 'Unisys terminal' should be interpreted to mean any TD, MT, ET, T27, or compatible display terminal device.

It is important to fully understand the emulation process BEFORE attempting to use the software. It is assumed the reader is familiar with the material presented in the following publications:

1. *Unisys MT 983/MT 985/MT 993 Display Terminal Systems:Reference Manual* (Form Number 1118361)
2. *IBM 3270 Information Display System: Data Stream Programmer's Reference* (GA23-0059-0)
3. *IBM 3179 G Color Graphics Display Station Description* (GA18-2261-0) or any other representative 3270 display station description

A list of related and useful publications is given below:

1. *IBM 3270 Information Display System Reference Summary* (Gx20-1878-4)
2. *Unisys ET2000 Series MT 985 Emulator* (Form Number: 1162294)

Structure of this manual

This document is divided into 8 major sections.

Chapter 1 Describes the manual itself and provides general information about the product.

Chapter 2 Gives installation details specific to the MT EMULATOR module.

Chapter 3 Describes the IBM 3270 display station.

Chapter 4 Describes the IBM 3270 data stream format.

Chapter 5 Describes the functionality of the MT EMULATOR.

Chapter 6 Describes the operation of the IBM 3270 display station under emulation.

Chapter 7 Describes the software interface to the MT EMULATOR.

Appendix A Lists the Unisys terminal Escape sequences.

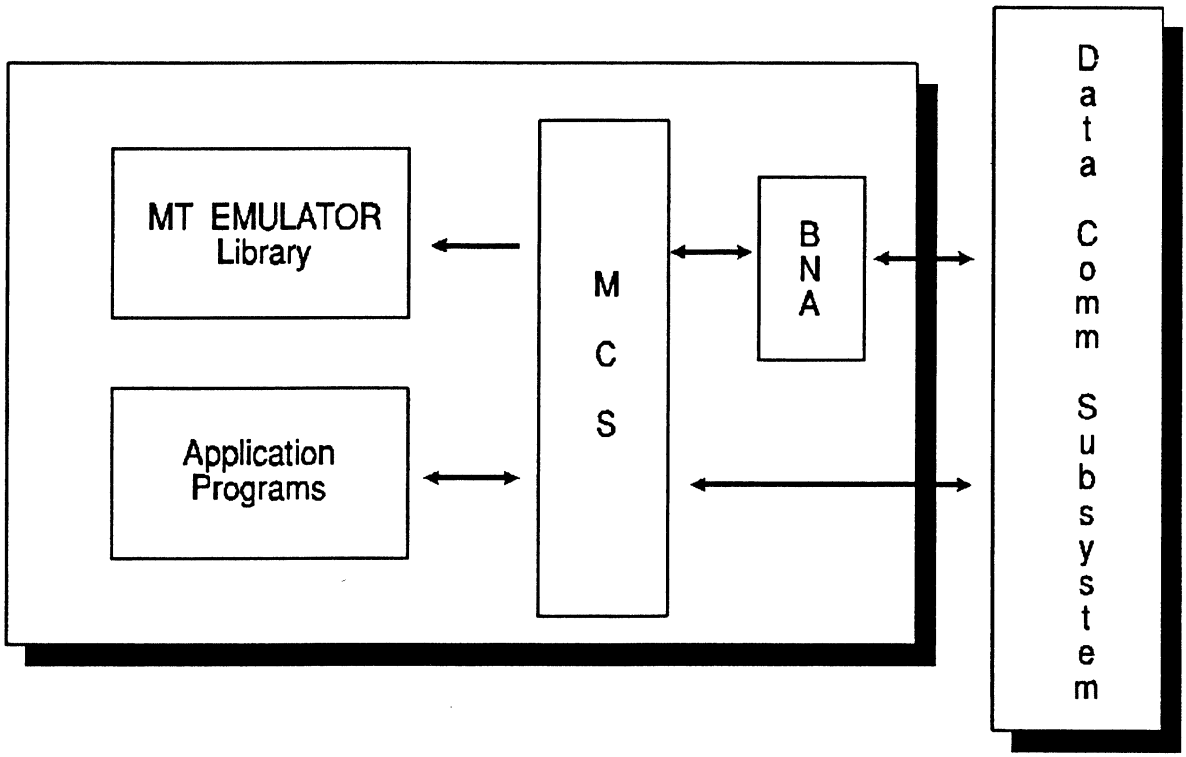
Appendix B Describes the Unisys terminal configuration.

Product Description

The MT EMULATOR is a software product that runs on Unisys Large Systems including the B5000/6000/7000 and A Series machines. This software product differs from other terminal emulators in one important respect, namely, the location of the MT EMULATOR software. This product runs on mainframes and not on terminals. For this reason, the MT EMULATOR software can only handle the details of data stream interpretation while lower level data comm protocol and network management details must be handled elsewhere. Therefore, the MT EMULATOR product is used along with a suitable Application Controller / Message Control System (MCS) and a suitable link level protocol.

More information on products using the MT EMULATOR can be found in Chapter 2, *Installation*.

Following is a system block diagram of the MT EMULATOR's operating environment:



Chapter 2
Installation

The MT EMULATOR is used with one of the following products:

- SNA ENVOY
- GATEWAY MESSAGE CONTROL SYSTEM
- COMS NETWORK INTERFACE III

These products will be referred to by the generic term, interface product, in the following discussion. Information provided in this section is in addition to the installation details provided with the interface product.

The MT EMULATOR is an independent library module and a separate code file. This module will always carry the MTEMULATOR name as the last level name. The object code file name along with the product name is listed below:

Code file name on release tape	**Product name**
ENVOY/OBJECT/LIBRARY/MTEMULATOR	SNA ENVOY
GATEWAY/IBMSUPPORT/OBJECT/LIBRARY/MTEMULATOR	GATEWAY MCS
COMS/OBJECT/LIBRARY/MTEMULATOR	COMS NI III

The MT EMULATOR software needs the services of a data management module to keep track of the terminal configuration information. This module is also employed by the ENVOY and GATEWAY products and thus can be shared.

The following is a quick reference list of modules used by the MT EMULATOR when functioning with the indicated products:

Product Name	**Module Name**	**Code File Name**
SNA ENVOY	MDM	SYSTEM/MDM
GATEWAY	GWSUPPORT	GATEWAY/NUCLEUS/OBJECT/LIBRARY/SUPPORT
COMS NI III	MDM	SYSTEM/MDM

MT EMULATOR Internal Files

Display Store File

This file holds the screen images for all the terminals. A new file is created every time the MT EMULATOR software is initiated. The file's name is constructed as follows:

```
<emulator code file's first level name>/MTEMULATOR/DISPLAYSTORE
```

For example, if the MT EMULATOR's code file name is ENVOY/OBJECT/LIBRARY/MTEMULATOR, then the Display Store File's name will be ENVOY/MTEMULATOR/DISPLAYSTORE.

The location of this file can be changed during configuration of the MT EMULATOR (see the section titled *Configuring the MT EMULATOR* later in this chapter).

Terminal Configuration File

This file stores the terminal settings that were defined for a station. If the MT EMULATOR is interfacing with SNA ENVOY, the Terminal Configuration File (TCF) is a logical file within the existing network configuration database. If the MT EMULATOR is interfacing with GATEWAY or COMS NI III, then the TCF is a separate physical file. The file is titled COMS/MTEMULATOR/CONFIGFILE in the COMS environment and GATEWAY/MTEMULATOR/CONFIGFILE in the GATEWAY environment. The TCF must be backed up regularly for recovery purposes.

Configuring The MT EMULATOR

The MT EMULATOR code file functions as a configurator when executed from a terminal logged onto the CANDE MCS. The interface product must not be running when the MT EMULATOR is executing in this mode.

1. Log onto CANDE. The MT EMULATOR code file as well as the Terminal Configuration Database must be available from the usercode.

2. Run the MT EMULATOR code file.

Home Menu

```
        MT EMULATOR Configurator Home Menu


[ ] Operational Parameters

[ ] Station Options

[ ] Key Programming(Not yet implemented)

[ ] EBCDIC Code Point Translation Chart

[ ] Quit Configurator


** Use SPCFY Key to make your selection **
```

This is the first screen that will be displayed when the MT EMULATOR is brought up in configuration mode. The function selected here will cause another screen to be displayed.

Operational Environment Screen

```
        MT EMULATOR Operational Environment Screen

Action: [  ]  HOme QUit MOdify

Code File Version: 13.03     Config File Format Level: 1

Update Capable User: YES    Usercode: [UPDATEUSER        ]

Location of Display Store File: [                  ]

Number of work buffers: [   ]

Maximum number of pages: [  ]
```

Update Capable User/Usercode

This information is not currently used.

Location of Display Store File

This option assigns the Display Store File to a particular pack. If the field contains only the slash character, /, the file will be allocated to memory. The default is the pack where the MT EMULATOR code file resides.

Number of work buffers

This option can be used in the multi-threading environments provided by the GATEWAY and COMS products to place an upper limit on the number of concurrent calls to the MT EMULATOR. When the number of tasks calling the MT EMULATOR at any given time reaches this limit, all subsequent calls will go into a waiting state until a buffer becomes available. The option is not applicable to the SNA ENVOY environment.

Maximum number of pages

This option controls the number of pages per terminal that the MT EMULATOR will support. This is a global option applying to all terminals; it cannot be set for each individual terminal. The number of pages must be in the range of two to sixteen, inclusive. The default is three. The number specified here will have a direct effect on the size of the Display Store File since an image of each page in use for each terminal is stored in this file.

EBCDIC Code Point Translation Chart

Some sites, especially non-U.S. sites, will have a need to define translate tables to provide support for local character codes. Sites with such a need can define these tables by selecting the EBCDIC Code Point Translation Chart option. The screen is presented below.

```
                EBCDIC Code Point Translation Chart

Action: [  ]  HOme QUit REfresh INquire CReate MOdify

Table Index: [  ]   Direction: [ ] Input/Output     Mode:[ ] N/E


   0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
0[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
1[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
2[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
3[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
4[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
5[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
6[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
7[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
8[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
9[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
A[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
B[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
C[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
D[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
E[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
F[  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ][  ]
```

Note: *The columns represent the high order digit of the hex code for the character and the rows represent the low order digit.*

Fifteen different translate tables can be defined. Each table is numbered and this number can be associated with a station in the Station Options screen. Each table provides for input (terminal to host) and output (host to terminal) translation. These "normal" translate tables do NOT affect the data stored in the MT EMULATOR screen buffer.

There is also an option to define extended code point translation. Characters following an Escape, shift-out code will be translated using the extended translation tables. The "extended" tables modify the data that is stored in the buffer. If both types of translate tables are defined, the extended tables are applied first on output to an IBM 3270 and applied last on input from an IBM 3270.

When using extended tables, it is also important not to introduce MT EMULATOR

sensitive control characters into the screen buffer by way of translation. Such operations can lead to unexpected results including MT EMULATOR faults.

An EBCDIC character is identified for translation by using its hex code to find the position on the screen representing the intersection of the high order digit with the low order digit. Enter into this position the hex code for the translated character.

Example

Convert the characters "[" and "]" between IBM EBCDIC and Unisys EBCDIC:

1. Enter *CR* in the Action field.
2. Enter a number in the Table Index field.
3. Enter *I* for input in the Direction field.
4. Enter *N* (normal/extended) in the Mode field.
5. Enter a *4A* in the field at column A and row D and a *5A* in the field at column B and row D.
6. Transmit the screen. This defines an input translate table.
7. The corresponding output table would be defined using the same procedure except that "MO" would be entered in step 1, "O" would be used in step 3, and the hex codes would be reversed in step 5.

On output, the hex code 4"FF" is reserved for the MT EMULATOR's use, and on input, 4"1E" is reserved.

Terminal Options

```
                MT EMULATOR Station Options

Action: [     ] HOme QUit REfresh INquire MOdify CReate ModifyALL

Station: [                  ]

EDIT:     [N] FULL FORMS PAGE XMIT      [N] LINE-AT-A-TIME XMIT
          [N] LOWERCASE ENABLED         [N] DC1 CLEARS TO EOL
          [N] DCOM LF -> CR             [Y] DCOM CR -> LF
          [N] DC2 ADV DCP               [N] VT PAGE ADV
          [N] VAR TABS                  [N] TAB RETAIN
          [N] FF CLR TABS

MISC:     [Y] STATUS DISPLAY ENABLED    [N] FIELD FILL ENABLED
          [N] OPTION "I" ENABLED        [Y] LSCANDE ENABLED

          [00] CODE POINT CHART

          ALTERNATE DELIMITER (HEX):    [00] OPEN  [00] CLOSE
```

The terminal options required for your application environment must be decided upon in cooperation with your site's system software support group. The MT EMULATOR's default terminal options, displayed on the screen image above, can be changed by creating a station called MODELMT and setting its options to the desired default values. When a station logs on, a search is performed for a record for that particular station name. If it does not exist, the MODELMT record is retrieved and those settings are used. If the MODELMT record does not exist, the default settings as displayed above are used.

Any changes made to the terminal options of MODELMT take effect immediately. For this reason MODELMT should not be the station name of any regular text editing or transaction processing station. Use of this station name must be restricted to setting up and modifying default terminal options of the MT EMULATOR.

The options on the screen are changed by entering a *Y* for yes and an *N* for no. Any character other than these two is interpreted as an *N*. If an invalid hex character is entered for the alternate delimiters, no change is made. If *00* is entered, the current delimiter settings are deleted.

It is extremely important to have the proper terminal options set up before attempting to use an IBM 3270 terminal for data retrieval and updates. Improper terminal options can produce undesirable side effects.

Unisys terminals make use of scratchpad memory areas for holding terminal configuration information. The MT EMULATOR stores this information in the Terminal Configuration File. The contents of the scratchpad can be modified by two methods: through a keyboard request or through a host request. (See Appendix B for a description of the Unisys terminal's scratchpad layout.) Not all Unisys terminal options are meaningful to the MT EMULATOR. Those that are, are described below. The default setting is in parentheses following the option name.

FULL FORMS PAGE XMIT (reset)

When this option is set, all unprotected and protected transmittable data throughout the screen is sent to the application program regardless of where the cursor was at the time of transmission. If this option is reset, only data up to the cursor position is transmitted to the program.

LINE AT A TIME XMIT (reset)

When this option is set, the transmit home location is at column one of the cursor line; otherwise the transmit home location is column one of line one.

LOWERCASE ENABLED (reset)

When this option is set, lowercase characters entered from the keyboard are NOT translated to uppercase characters. Setting or resetting this option will have no effect if the code point chart in use for this station is other than chart 0.

DC1 CLEARS TO EOL (reset)

When set, all characters from the data comm cursor to the end of the line or field, depending on forms mode, are replaced with blanks.

DCOM LF -> CR (reset)

When set, a carriage return action is performed after doing a line feed.

DCOM CR -> LF (set)

When set, a carriage return moves the data comm cursor to column one of the next line; otherwise the data comm cursor moves to column one of the same line.

DC2 ADV DCP (reset)

When set, receipt of a DC2 character moves the data comm cursor to the left. If this option is reset, receipt of the DC2 character complements the forms mode state of the data comm page.

VT PAGE ADV (reset)

When set, receipt of a VT character invokes vertical tab down. If this option is reset, the VT character is ignored.

VAR TABS (reset)

Not implemented.

TAB RETAIN (reset)

Not implemented.

FF CLR TABS (reset)

Not implemented.

STATUS DISPLAY ENABLED (set)

When set, system status information like the page number, presence of an application status line, etc. is displayed in the bottom right-hand corner of the display device.

FIELD FILL ENABLED (reset)

When set, alphanumeric fields will be padded with the character "_" instead of blanks, and numeric fields will be padded with "*".

OPTION 'I' ENABLED (reset)

When set, the screen is displayed as if it were under control of a "?+I". When reset, the screen is displayed as if it were under control of a "?-I".

LSCANDE ENABLED (set)

When set, the terminal is assumed to be an LSCANDE virtual terminal interfacing to Large System applications. When reset, the terminal is assumed to be translucent, sending and receiving all characters, including carriage control.

Note: *Users accessing Medium Systems applications must reset this option.*

CODE POINT CHART (00)

A value in the range of 1 to 15, is considered an index into a set of translate tables. These translate tables are used for translating data streams on input and output.

OPEN ALTERNATE DELIMITER (00)

Specifies a character in hex that is replaced by a unit separator (US) character. The value 00 resets the alternate delimiter character.

CLOSE ALTERNATE DELIMITER (00)

Specifies a character in hex that is replaced by a record separator (RS) character. The value 00 resets the alternate delimiter character.

Resource Requirements

Approximate memory requirements for the MT EMULATOR with no active sessions are shown in the table below. Each active session will add four words of memory.

	Save	Total
Stack	1942	4854
Code	273	17465
Total	2215	22319

The location of the Display Store File, in memory or on disk, can have a major effect on the memory requirements quoted above. The file's size depends on the number of active MT EMULATOR sessions and the number of display pages in use for those sessions. If S is the total number of active sessions and P is the average number of pages in use for those sessions, 13 * S * P sectors or 2340 * S * P bytes will be the average requirement.

Chapter 3
IBM 3270 Display Station

This section provides general information on the IBM 3270 derived from the IBM 3270 display station description [3]. For information regarding the specific device in your site, you must consult the appropriate device manual.

The IBM 3270 display station basically consists of three elements: the video element, the logic element and the keyboard.

The video element consists of a cathode-ray tube with different screen and character sizes depending on the model. For the proper functioning of the MT EMULATOR, the model must support a screen capable of displaying 1920 characters in a 24 lines by 80 characters wide format. The bottom of the screen is reserved for displaying status information. A representative IBM 3270 display station screen is shown in Figure 3-1.

The screen is shown illustrating the cursor and the operator information area. The operator information area is used exclusively by the display device and the controller associated with the device.

The logic element contains the logic that controls the functions of the display device. The logic element is connected to an IBM 3274 style control unit.

The keyboard element can be as varied as the video element. A popular type of keyboard is the 122-key typewriter keyboard. The keyboard is connected to the logic element.

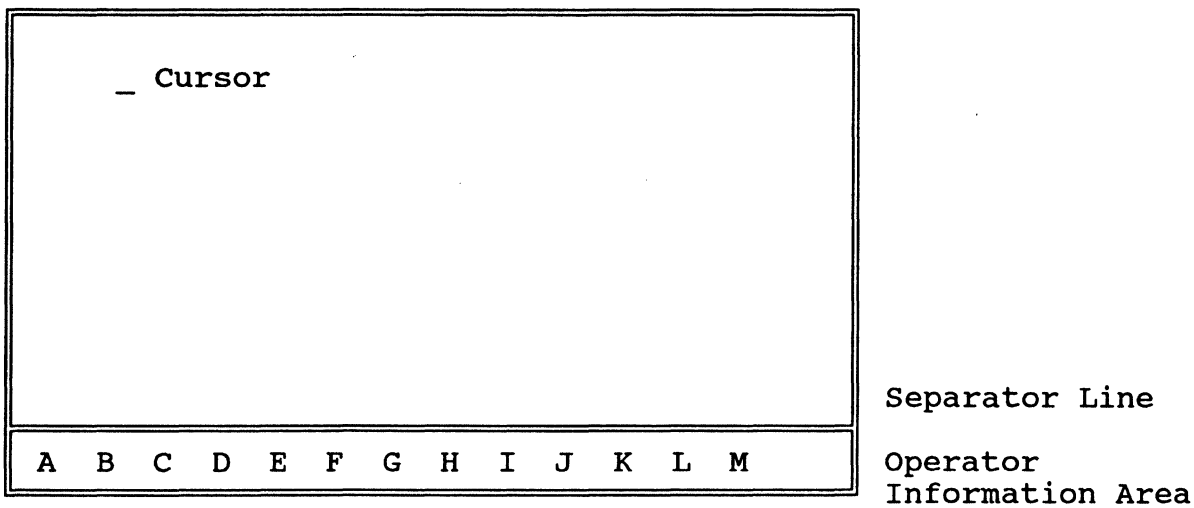


Figure 3-1: IBM 3270 Display Station Screen

IBM 3270 Video

This section describes the layout of the operator information area and some of the symbols that will appear there. For a detailed discussion, the reader must study the appropriate device documentation.

The operator messages can be categorized into the following groups which appear at the designated positions in the information area:

Area	Description
A	Readiness and system connection
B	Do-Not-Enter conditions
C	Communication error conditions
D	Background transparency selection
E	Program symbol selection
F	Character selection
G	Shift selection
H	Highlight selection
I	Extended color selection
J	Mode symbol
K	Printer status
L	Entry assist
M	Cursor position

Not all areas are used by all the devices. For example, area I would not be applicable to a monochrome video element.

A brief discussion of some of the operator messages appearing on the IBM 3270 display device follows.

Readiness and System Connection

This area shows the operative and online status of the device.

	This symbol indicates that the device is operative and under the control of the associated controller hardware.
A or **B**	Indicates that the display device is connected to the host under Non-SNA or SNA rules, respectively.
	When solid, this block indicates that the device is connected to a task. In an SNA environment, this implies that the device is owned by a LU-LU session.

Do-Not-Enter

The Do-Not-Enter (X) indicator informs the operator that the device will not accept input from the keyboard. Symbols to the right of the "X" indicate the reason for disabling input. As long as the "X" is displayed, the keyboard is logically locked and no function can be performed. Some of the Do-Not-Enter conditions can be removed by pressing the Reset key.

Following is a list of some of the Do-Not-Enter conditions:

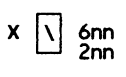
Terminal Wait

The "X" symbol followed by the clock symbol informs the operator that the host is processing the operator's request. You must wait for the function to be completed.

Go Elsewhere

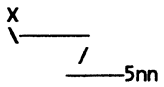
The "X" symbol followed by a stick man symbol flanked by arrows on either side indicates that an attempt was made to enter data in a protected field or on top of an attribute byte. This condition can be removed by pressing the Reset key.

Machine Check



This symbol indicates the display station is not working correctly.

Communication Check



This symbol indicates there is a problem with the communication line to the host system.

Program Check

XPROG7nn
XPROG4nn

These messages indicate the display station detected a programming error in the data from the host processor. This may appear because of an incompatibility between the application program and the display device. Error codes starting with a four are issued by the controller. Pressing the Reset key will remove this indication. If this condition persists, contact product support.

What ?

X?+	This symbol indicates the last operation was not accepted by the display device. Usually this is the result of trying an operation when a Do-Not-Enter condition is present. Press the Reset key to remove this condition.

System Lock

XSYSTEM	This message indicates that the application program has disabled the keyboard and data cannot be entered. The Reset key can remove this indication and unlock the keyboard.

For more information on operator messages and for an explanation of the error codes, the reader is referred to the display station manual for the specific device in question.

IBM 3270 Keyboard Functions

This section describes a representative IBM 3270 display station's keyboard. The layout of the keys is model-dependent. Only those key functions meaningful to the MT EMULATOR are discussed in this section.

Left Control Keys

This group consists of ten keys located on the left-hand side of the keyboard.

Clear Key

When accepted by the display device, this key:

- Erases the screen
- Moves the cursor to row one, column one of the screen
- Transmits a CLEAR AID to the host

Erase EOF Key

This key erases all the characters from the cursor position to the end of the field while the cursor is located in an unprotected field. If the screen has no fields defined, all characters from the cursor position to the end of the screen will be erased.

ErInp Key

The ErInp key can be used to erase the contents of all unprotected fields in the screen and to move the cursor to the first position of the first unprotected field. If the screen is unformatted (has no fields defined), the entire screen is erased and the cursor moves to column one of row one. If the screen has only protected fields, no data is erased, but the cursor moves to column one of row one.

This key will not function properly with the MT EMULATOR. Use a programmed key to emulate this key.

Selectable Field Tab (->>|) Key

This key can be used to tab to the first character position of the next cursor-key-detectable field with a designator character (unprotected or protected).

Right Control Keys

This group consists of 10 keys located between the data keys and the numeric keypad.

Cursor Left (<-) and Cursor Right (->) Keys

These keys can be used to move the cursor one position to the left or to the right. The cursor wraps across lines and the end of the screen.

Cursor Up and Down Keys

These keys are similar to the key group above, but now the cursor moves one position either up or down.

Cursor Home Key

If the screen contains unprotected fields, pressing this key moves the cursor to the first position of the first unprotected field. In all other cases, the cursor moves to column one of row one.

Back Tab (|<-) Key

This key can be used to move the cursor to the first position of an unprotected field, provided the cursor is already located in that field.

Delete Character Key

This key has the symbol of an 'a' with a line across it. Pressing this key deletes the character at the cursor position if the cursor is located in an unprotected field or in an unformatted screen. All characters in the field to the right of the cursor and on the same line move one position to the left. A null character is inserted after the last character in that field.

If the cursor is located in a protected field or on top of an attribute character when this key is pressed, a 'go elsewhere' symbol appears in the operator information area.

Insert Mode Key

This key has an 'a' symbol with a '^' symbol on top of it. It places the keyboard in insert mode and a '^' symbol appears in the operator information area. If there is at least one null character between the cursor and the end of the field, typing a character in an unprotected field causes the following:

- The character at the cursor and all characters to the right of the cursor, up to but not including the first null character, move one position to the right.

- The keyed character is displayed at the cursor position.
- The cursor advances to the next character position.

This insert operation wraps across lines if the field under question spans multiple lines.

If the field contains no null characters, an attempt to insert a character causes a Do-Not-Enter symbol to appear in the operator information area. Pressing the Reset key will remove this condition.

The following keys terminate insert mode:

```
CLEAR, ENTER, PA1-3, PF1-24
```

Program Access Keys

The program access keys are meaningful only to the application program. The MT EMULATOR assumes this role. Pressing a PA key while an inbound data stream is being processed causes the Do-Not-Enter symbol "X?+" to appear. Press the Reset key to remove this condition.

Program Function Keys

The program function key group consists of 24 keys located above the data keys. When pressed, each transmits a unique AID code to the host. Pressing a PF key while an inbound transmission is being processed causes the Do-Not-Enter "X?+" symbol to appear. Press the Reset key to remove this condition.

Data Keys

This key group is located between the left and the right control keys. This group consists of alphabetic keys, numeric keys, punctuation character keys, New Line key, Reset key and the Enter key. A useful addition to this group of keys is the Tab key, Backspace key and the Shift Lock key.

Backspace Key

This key moves the cursor one position to the left. In insert mode, this key will delete the character to the left of the cursor position moving the cursor one position to the left. All characters to the right of the deleted character on the same line within the same field shift by one position.

If the backspace key is pressed in insert mode and the character to the left of the cursor is an attribute, or the cursor is in a protected field, a Do-Not-Enter symbol appears. Press the Reset key to remove this condition.

Alphabetic Keys

The characters on the face of the keys are displayed in lowercase, unless used with the Shift or the Shift Lock key.

Numeric and Punctuation Mark Keys

The numeric and the punctuation mark keys are provided on the same keys. The symbols appearing on the upper portion of the key can be entered by holding down the Shift key and the desired symbol key.

Enter Key

Pressing this key transmits data from the screen to the application program.

Newline (<-|) Key

This key moves the cursor to the first unprotected character position in the line following the current line.

Reset Key

The Reset key:

- Causes the keyboard to leave insert mode
- Unlocks the keyboard and removes Do-Not-Enter conditions

Tab (->|) Key

The Tab key moves the cursor from the current position to the next unprotected field. If the screen is unformatted, the cursor moves to column one of row one.

Shift Lock Key

This key locks the keyboard in the uppercase shift condition. Press the Shift key to get back to the lowercase mode.

Shift Key

Use this upward pointing arrow key to obtain the symbols on the top line of the multiple symbol data keys.

Numeric Key Pad

This group is located at the extreme right of the keyboard. It consists of the numeric (0-9) keys and the Space, Tab and Enter keys.

Chapter 4
IBM 3270 Data Stream

The IBM 3270 data stream format is designed primarily for screen presentation, but it also supports printer data. Each character position on the screen has a matching storage buffer address in the IBM 3270 cluster controller providing a one-to-one correspondence.

The outbound data stream (application to terminal) is composed of a one-byte command, a one-byte control character (WCC) used only in write commands, and the data. The inbound data stream (terminal to application) is composed of a one-byte AID (Attention Identifier), a cursor position, and the data. The data portion in both the outbound and inbound streams can contain "orders" and "attributes" to control the formatting of the text.

Commands

Commands are used by applications to read or write to display terminals. A command is always the first character in the outbound data stream.

Command	EBCDIC
Write	F1
Erase Write	F5
Erase Write Alternate	7E
Read	F2
Read Modified	F6
Read Modified All	6E
Erase All Unprotected	6F

Figure 4-1: IBM 3270 Commands

WRITE Command

The WRITE command applies data stream modifications to the current buffer using the current buffer address and current data information. The data and orders within the data stream will modify information within the character buffer.

ERASE WRITE Command

The ERASE WRITE command will first clear the character buffer of all information and fill it with nulls, set the current buffer address to zero, and then perform the operations of modifying the character buffer with data and orders.

ERASE WRITE ALTERNATIVE

The ERASE WRITE ALTERNATIVE command functions the same as the ERASE WRITE command but it uses the alternate partition size.

READ BUFFER

READ BUFFER causes the entire buffer contents of the addressed terminal to be read into main storage.

READ MODIFIED BUFFER

READ MODIFIED BUFFER sends all fields modified by the keyboard to the application.

READ MODIFIED ALL

READ MODIFIED ALL is the same as READ MODIFIED BUFFER except that it sends data on any AID type.

Write Control Character

The Write Control Character (WCC) always follows a WRITE, ERASE WRITE, or ERASE WRITE ALTERNATE command. Each bit in the character controls a specific function. The layout is as follows:

BIT	Explanation
0	Determine by contents of bits 2 through 7
1	WCC reset bit
2, 3	Define the printout format: 00 = Use print orders in data stream for formatting 01 = 40 character print line 10 = 64 character print line 11 = 80 character print line
4	Start printer bit. When set to 1, indicates a printout operation at the completion of the write operation.
5	Sound alarm bit. When set to 1, sounds alarm on selected device.
6	Restore Keyboard bit. When set to 1, restores operation of keyboard.
7	Reset MDT bit. When set to 1, resets all Modified Data Tags in the selected device buffer before executing any orders or writing data.

Figure 4-2: WCC Description

Note: *The bit layout is from the most significant (0) to the least significant (7).*

If no information follows the WCC character the command is completed. This type of write command is used to restore the keyboard to the user without changing any data. If data and/or orders follow the WCC command the character buffer is modified accordingly.

Attention Identifier (AID)

An AID is the first character of an inbound IBM 3270 data stream. This data stream is initiated by the execution of a READ command which could have been issued by an application or caused by an operator action at the keyboard. The AID character indicates the type of action. Following is a list of AIDs:

AID	EBCDIC (HEX)	AID	EBCDIC (HEX)
No AID	60	PF13	C1
Structured Fields	88	PF14	C2
Read Partition	61	PF15	C3
Test Req and Sys Req	F0	PF16	C4
PF1	F1	PF17	C5
PF2	F2	PF18	C6
PF3	F3	PF19	C7
PF4	F4	PF20	C8
PF5	F5	PA1	6C
PF6	F6	PA2	6E
PF7	F7	PA3	6B
PF8	F8	Clear Key	6D
PF9	F9	Clear Partition	6A
PF10	7A	Enter Key	7D
PF11	7B	Selector Pen	7E
PF12	7C	Magnetic Reader	E7

Figure 4-3: Attention Identifiers

Once an AID is set by the terminal, it remains set and input is inhibited until a WRITE command with the keyboard restore bit on is received or an ERASE ALL UNPROTECTED command is received.

Orders

Orders can be present in both the outbound and inbound data streams. They are unique one-byte control characters and can occur anywhere in the data. Two types of orders exist: buffer control orders and print format orders. Buffer control orders are interpreted and executed as they are received by the control unit, and are used to position, define, modify, assign attributes on a field and character basis, and format data being written to a display character buffer; to erase selected unprotected data in the buffer; and to reposition the cursor. Printer formatting orders are initially stored in the printer character buffer as data, and are interpreted and executed by the printer logic when encountered in the print operation.

Order	Code	Definition
SF	1D	Start Field; identifies the next character as the Attribute Character for this field.
SBA	11	Set Buffer Address
IC	13	Insert Character
PT	05	Program Tab
RA	3C	Repeat to Address
EUA	12	Erase Unprotected to Address

Figure 4-4: IBM 3270 Order Codes

Set Buffer Address

The Set Buffer Address order is followed by two address characters, and sets a pointer called the Current Buffer Address (CBA). Subsequent data characters will be stored in the character buffer in sequential fashion beginning at the CBA. As each character is stored the CBA is updated to point to the next character location in the character buffer.

Start Field

The Start Field order indicates that the next character in the data stream is to be interpreted as a field attribute character. It will therefore be stored in the character buffer in a unique fashion so that the hardware will interpret it as a field attribute and not as a data character.

Insert Cursor

The Insert Cursor order causes the cursor to be displayed at the screen location associated with the CBA.

Program Tab

The Program Tab order will cause the CBA to be set to the first data character position in the next unprotected field in the buffer. Under certain circumstances it will also cause nulls to be inserted into the character buffer from the new CBA to the end of the field.

Repeat to Address

The Repeat to Address order is followed by two character buffer address bytes and a character to be repeated. That character will be replicated through the character buffer up to, but not including, the specified buffer address.

Erase Unprotected to Address

The Erase Unprotected to Address order is followed by two character buffer address bytes and causes nulls to be inserted in all unprotected buffer locations starting at the CBA and up to, but not including, the specified stop address.

Field Attributes

A one-character attribute follows the Start Field order. It defines the beginning of a field and all the characteristics of that field. It takes up one character position, and is displayed on the screen as a space. The end of the field occurs when the next Start Field order is encountered.

Bit Position	Value	Definition
Bit 0	0/1	Make a graphic character
Bit 1	1	Unconditional
Bit 2	0 1	Unprotected Protected
Bit 3	0 1	Alphanumeric Numeric
Bit 4 & 5	00 01 10 11	Normal Display Normal Display Intensified Display Secured - Not Displayed
Bit 6	0	Unconditional
Bit 7	0 1	Field not Modified Field Modified

Figure 4-5: Field Attribute Layout

Note: *Attribute Byte Layout and Definition Bits are numbered from MSD (0) to LSD (7) positions.*

Field attributes can only be overwritten by the keyboard operator with the Clear key. This operation nullifies the entire screen, including all attributes. An application program can overwrite field attributes with data.

Protected/Unprotected

An operator cannot enter data into or modify the content of a protected field. Input fields that require data from the operator must be unprotected.

Alphanumeric/Numeric

In an unprotected input field, alphanumeric/numeric defines the type of data that an operator can enter into the field. This attribute has special meaning for protected fields.

Non-display, Display, and Intensified

Data contained in the field is either not displayed, displayed at normal intensity, or displayed at high intensity.

Modified Data Tag

Field attributes are protected against input from the keyboard. However, bit seven (Modified Data Tag) is set to one when the operator enters data into the field defined by the attribute. Attribute characters are not protected against operation of the Clear key.

Chapter 5
MT EMULATOR Functional Description

Unisys terminal emulation is performed by this software at the mainframe level, not at the device level. This chapter describes the functional details of this emulation.

It is assumed the reader has a working knowledge of Unisys terminals and understands the material in the *Unisys MT 983/MT 985/MT 993 Display Terminal Systems: Reference Manual.*

Keyboard and Data Comm Cursor

The IBM 3270 terminal device has two sources of input, the host application and the keyboard. Each has a cursor denoting the position on the screen where the incoming data is to be stored.

The keyboard cursor changes as the operator enters data. Since the MT EMULATOR runs on the mainframe, all changes to the screen will be registered in the MT EMULATOR display buffer only *after* the operator transmits the data. The keyboard cursor in the MT EMULATOR is updated only at this time.

The data comm cursor is controlled by the host application or by specific operations that align the two cursors.

The position of these two cursors are entirely independent. They can be in different lines or even in different pages.

Display Buffer

The data to be displayed on the terminal is stored in the display buffer of the MT EMULATOR. The buffer receives input either from the terminal or from an application program. It is also the source of information to be transmitted to an application program. The display buffer is divided into lines of 80 characters and pages of 24 lines.

Terminal State

Each terminal has state information associated with it. This information is kept in association with the data comm and keyboard cursor locations. For example, the data comm state information consists of the page the data comm cursor is on, if this page is in forms mode, and the address of the cursor location. The keyboard state information is the same.

In addition to this, information is kept to indicate whether the terminal is off-line and, if so, for what reason. A terminal is switched to an off-line state for the duration of the following operations:

- Terminal configuration
- Keyboard emulation
- Displaying application status line
- Displaying product status information

In other words, whenever a terminal is conversing with the MT EMULATOR, it is considered to be off-line. A terminal in an off-line state can not receive data from any application program.

Video Character Set

IBM 3270 information display systems do not support the entire graphic set available with the Unisys terminals. For this reason a translate table is provided in the MT EMULATOR to allow all alphanumeric (upper as well as lowercase), numeric and a selected set of special characters to pass through while converting the remaining characters to question marks.

Video Highlights

Only bright and secure highlights are recognized by the MT EMULATOR. The remaining highlight characters, underline, reverse video, and blinking, are converted to a bright character. Since highlighting characters are only allowed with field definitions, any bright character appearing within an input field is skipped while bright characters appearing elsewhere are processed.

Application Status Line

The application status line is a data structure consisting of 73 or 78 characters, depending on the setting of STATUS DISPLAY, whose contents are provided by the host application. Since the IBM 3270 device has no such concept, the application status line is implemented as follows:

The application status line resides in the same buffer page as the data comm page. The terminal state table keeps track of this information. There is only one application status line per terminal.

If the 24th line of the buffer page can accommodate the status line message without overwriting any information on that line, the status line is inserted there. If STATUS DISPLAY is disabled, a maximum of 78 characters will be displayed. If STATUS DISPLAY is enabled, a maximum of 73 characters will be displayed.

If the 24th line cannot accommodate the status line message and the Status Display station option is enabled, then the characters "msg" are displayed in the system status area of the screen indicating that an application status message exists. (This indication is given only if the keyboard page and the application status page are the same.) The information in the application status line can be viewed at any time by pressing a function key.

The application status line is forgotten when the Clear key is pressed. Typical use may involve viewing the status line and then pressing the Clear key. The status line information will continue to appear until the Clear key is pressed or the keyboard page is different from the page where the application status line is stored.

System Status Line

This status line is used to display system status indications. Like the application status line, this feature does not exist on the IBM 3270 device and so is implemented as follows:

The system status area is the last six-character position of the display surface. System status messages may or may not be displayed depending on the setting of the terminal option Display Status. If enabled, status messages are displayed only if they do not overwrite data in that area.

The information displayed in the system status area includes the terminal page number, "pg n"; the application status line indication, "msg"; and an indication when data is being received on a page different from the keyboard page, "rcv".

Page Formatting

Forms Mode

By default, every page is a non-forms page. A page can only be put into forms mode through an explicit program request. The exception to this rule is the special CANDE page feature. (See CANDE Forms later in this chapter.) The Escape-W control key sequence will initiate a forms mode request, but the page is considered a forms page only if the page has at least one field defined.

Data streams directed to Unisys terminals can request four types of fields: left-justified, right-justified, protected transmittable, and fixed output. IBM 3270's provide for only three types of fields: unprotected alphanumeric, unprotected numeric, and protected. These fields end only when another field definition is encountered. There is no counterpart for the Unisys terminal end field delimiter in the IBM 3270 data stream.

Left-Justified Field

On the Unisys terminal this field is delimited by a start field "US" character and an end field "RS" character. On the IBM 3270 the "US" character position becomes the attribute byte which marks the field as alphanumeric. The end field character becomes a protected field attribute byte, thereby ending the field at this point.

Right-Justified Field

On the Unisys terminal this field is delimited by the "GS" start field character and the "RS" end field character. Processing is identical to the left-justified field case, except that the attribute byte now marks the field as numeric.

Protected Transmittable Field

On the Unisys terminal this field is delimited by the "FS" and "RS" characters. The start field character position becomes the attribute byte on the IBM 3270 which marks this field as protected. This field will not be transmitted back to the MT EMULATOR by the IBM 3270, unlike the Unisys terminal. Information in this field is actually reconstructed by the MT EMULATOR before sending the data to the application program.

Fixed Output Field

On the Unisys terminal a fixed field is any part of the form that is between an end field character and a start field character. Data in this area will not become part of the information going back to the application. Fixed output fields are protected fields on the IBM 3270.

CANDE Forms

A special screen is displayed on the IBM 3270 to handle CANDE pages which causes each line on the screen to be represented as a field. The last character of each line is an attribute byte specifying an alphanumeric field. This approach reduces the amount of data transmitted to the MT EMULATOR, since only modified fields, in our case modified lines, are transmitted. A side effect is that the last character position of each line is not available for display or data entry. However, the data in the last character position of each line is intact in the MT EMULATOR buffer and so it is possible to use CANDE page mode to modify displayed data.

Non-Forms Page

By default, every page is a non-forms page. An Escape-W control sequence will change the page to forms mode unless no fields are defined. The Escape-X control sequence makes the data comm cursor page a non-forms page unconditionally. A non-forms screen will be completely protected when it appears on the IBM 3270 display surface. In order to enter any data on the screen, the operator must clear the screen. These screens can be used to prevent certain IBM 3270 controllers from crashing when presented with data from a terminal exceeding a certain size.

Terminal Operations Controlled via Data Comm

The terminal operations available through the keyboard are discussed in Chapter 6. The functions that can be requested by application programs are discussed below. Appendix A lists the control code sequences for these functions.

Text Editing

Line Manipulation Group

Line Insert

Inserts a blank line at the cursor row, deletes the last line of the page, and shifts the remaining lines down one row. The cursor is left at column one of the same line. This function is not available in forms mode.

Line Delete

Deletes the cursor row, inserts a blank line at the bottom of the page, and shifts the remaining lines up one row. The cursor is left at column one of the same line. This function is not available in forms mode.

Move Line Up

Exchanges the cursor line with the line above. The cursor remains in the same column one line above. If the cursor line is the top line of the page, it is exchanged with the bottom line of the page. This function is not available in forms mode.

Move Line Down

Exchanges the cursor line with the line below. The cursor remains in the same column one line below. If the cursor line is the bottom line of the page, it is exchanged with the top line of the page. This function is not available in forms mode.

Clear to End of Line

In a non-forms page, the clear to end of line order replaces all characters from the cursor to the end of the line with blanks. In a forms page, clear to end of line is operative only if the cursor is in a field delimited by start and end field characters. In this case, if the cursor is in a left-justified field, all characters from the cursor to the end field delimiter are replaced with blanks, while in the case of a right-justified field, the entire field is filled with blanks, regardless of the cursor position within the field.

Page Manipulation Group

Clear Page

In a forms page, the clear page order clears all left- and right-justified data fields, while in a non-forms page all data on the page is cleared.

Clear to EOP

In a forms page, the clear to end of page order first executes a clear to end of line, and then continues to blank all subsequent left- and right-justified fields. In a non-forms page, all character positions from the cursor to the end of page are filled with blanks.

Roll Page

Roll page up shifts the page up one line, moving the first line of the page to the bottom of the page. Roll page down shifts the page down one line, moving the last line to the top of the page. After the operation, the cursor remains on the same page, row, and column. This function is not available in forms mode.

Cursor Manipulation Group

Move Cursor

Move cursor up, down, left, or right affects the data comm cursor, depending on terminal options.

Home Cursor

Home cursor moves the data comm cursor to the first line and first column in the case of a non-forms page. Within a forms page, the same order moves the data comm cursor to the first position of a left- or right-justified data field.

Move Cursor to Column, Row

This moves the cursor to the specified column and row.

Page 'n'

Jump to page 'n' moves the data comm cursor to the home position of the requested page. If the requested page is greater than the maximum page allowed by the MT EMULATOR, the page jumped to defaults to the maximum page.

Tabs

Currently only fixed tabs are supported. In a non-forms page, the tabs are fixed at every eighth column or row starting with the first position, i.e., 1, 9, 17, etc. In a forms page, horizontal tab moves the data comm cursor to the first position of the next left- or right-justified data field.

Terminal Control Functions

The following terminal control functions are supported by the MT EMULATOR:

Transmit Firmware Version

This command is intercepted by the MT EMULATOR and its version information is sent back to the calling program as an EBCDIC string. The remaining data stream is ignored.

Lowercase

The lowercase command enables or disables the lowercase terminal option temporarily.

Transmit Page

The transmit page to host command is controlled by the MT EMULATOR. The return message to the host is generated by the MT EMULATOR. The MT EMULATOR expects an NDL style "?" command.

Search Mode

Search mode commands are not supported by the MT EMULATOR.

Video Control

Video control escape sequences, such as making the data comm page a reverse video page and so forth, are not supported by the MT EMULATOR.

Peripheral Control Commands

Peripheral control escape sequences, such as printing the whole page with form feed and so forth, are not supported by the MT EMULATOR.

Key Programming

The Escape-R-K control sequence is not supported by the MT EMULATOR.

Terminal Configuration

Terminal options can be set when running the MT EMULATOR in configuration mode (see Chapter 2), when executing real-time (see Chapter 6) and by control sequences received in the input message. The following Escape sequences in the input message cause the terminal options to be read and/or changed.

ESC R A hhhh kk <ASCII string>

This command replaces the contents of the scratchpad addresses hhhh through hhhh + kk - 1 with the ASCII string.

ESC R H hhhh kk <hex pairs>

This command interprets the next kk bytes as hex pairs represented in EBCDIC. The contents of the scratchpad addresses hhhh through hhhh + kk - 1 are altered.

ESC R C

This command essentially has no effect on the configuration. Normally this follows an ESC-R-H or an ESC-R-A sequence which would have already altered the scratchpads. This command clears all pages and sends out six asterisks.

ESC R P

This command writes the current scratchpad contents to the terminal configuration file, thereby making the changes permanent. Without this command, any changes made to the scratchpad are considered temporary and thus forgotten after a normal or an abnormal termination of the MT EMULATOR software. All pages are cleared and six asterisks appear on the display surface.

ESC R T hhhh kk

This command transmits the contents of the scratchpad, starting at address hhhh and ending at address hhhh + kk - 1, back to the application program.

ESC <space> D

This command initializes the terminal with the scratchpad contents restored from the terminal configuration file. All pages are cleared and six asterisks appear on the display surface.

NDL II Editing of Data Stream

The following discussion is only applicable to those terminals that have the LSCANDE station option enabled. It assumes the reader is familiar with the NDL II terminology and its logical-io editors and adapter control sections.

Data streams directed to the terminal network in a Unisys environment are massaged by the NDL II editor before reaching the display device. In order to provide a similar effect, the MT EMULATOR has the provision for applying an NDL II style output editing to the data stream before interpreting it. The interface calling the MT EMULATOR can request output editing and, in that case, the interface must supply the carriage control and the toggles information to the MT EMULATOR. The editor used by the MT EMULATOR is logic compatible with the standard NDL II logical-io output editor. This editor also appends a carriage return and erase to end of line commands to the front of the data stream if the output to the terminal does not start with an Escape character and is after an input from the terminal. This logic is the counter part of the logic in the input adapter control process of logical-io. Any data stream starting with an Escape, a DC1, a DC4, or a form feed character will not be edited.

Since the MT EMULATOR supports NDL II style output editing, it can also support scrolling, if required. ?+S, and ?-S commands can be issued from terminals to enable or disable scrolling. ?+I and ?-I commands are also supported. These are the only "?" commands supported by the MT EMULATOR at this time.

Chapter 6
IBM 3270 in Emulation Mode

The description of the IBM 3270 display device is covered in Chapter 3. Since the IBM 3270 does not address all the functions that can be performed on a Unisys terminal, logic has been provided in the MT EMULATOR to perform some of these functions.

Invoking Unisys Terminal Functions

MT EMULATOR logic to reproduce specific Unisys terminal functions is executed when the MT EMULATOR sees a particular control sequence in the input data. This control sequence is inserted via one of the IBM 3270's program function (PF) keys or one of the program attention (PA) keys. These keys are programmed to emit a particular control code sequence which will in turn execute the function in the MT EMULATOR. The keys are initialized with a default set of control code sequences, but they can be reprogrammed by the local site to reflect their environment. This key programming is done for the entire network, not on a terminal by terminal basis.

The list of MT EMULATOR control code sequences and the Unisys terminal function that they perform is displayed below and on the next two pages:

Emulator Control Code (HEX)	Unisys Terminal Code	Function - performed against the buffer stored in the MT EMULATOR unless specified otherwise
A0	Esc &	Align display cursor to data comm cursor
FC		Displays previous terminal page
D5		Moves the cursor back one character in the buffer
80	0C (HEX)	Clears the page and moves the cursor to home
86	Esc K	Clear to end of line
87	Esc J	Clear to end of page
E7		Enter configuration mode (See the Real-Time Configuration section later in this chapter.)
A4	Ctrl	See the next table for the valid Ctrl sequences

Figure 6-1: MT EMULATOR Control Codes

Emulator Control Code (HEX)	Unisys Terminal Code	Function - performed against the buffer stored in the MT EMULATOR unless specified otherwise
F9		Moves cursor down one line in the buffer
F6		Moves cursor up one line in the buffer
F8		Moves cursor right one character in the buffer
F7		Moves cursor left one character in the buffer
A2	03 (HEX)	End of text
AE	3C (HEX)	Moves cursor to home
85	Esc M	Deletes one line at keyboard cursor location (not performed if in forms mode)
83	Esc L	Inserts one line at keyboard cursor location (not performed if in forms mode)
A9	Local key	Clears the application status line message (must be the only data in the input message)
FE		Displays next terminal page
D6	0D (HEX)	Carriage return
C4		Moves cursor to previous field (only performed if in forms mode)
FA	Esc T	Roll Down
FB	Esc S	Roll Up
A6	Spcfy key	Transmits cursor position to application (must be the only data in the input message)
C6		Moves cursor to next field (only performed if in forms mode)
AC	XMIT	

Figure 6-1 Continued: MT EMULATOR Control Codes

The functions that have been implemented for the Ctrl A4 code (in the previous table) are in the following table. The characters following the A4 in the table below are coded in ASCII hex.

Emulator Control Code (HEX)	Unisys Terminal Code	Function - inserts the Unisys terminal code in the destination buffer unless specified otherwise
A42044A4	Esc space D	Initialize terminal options (not inserted in destination buffer)
A42056A4	Esc space V	Retrieves the MT EMULATOR version number
A442	Esc <	Exchange cursor line with line below
A4483A3F (HEX)		Begin bright video field
A4482F0F (HEX)		Begin underline video field
A4482E0E (HEX)		Begin reverse video field
A4483818 (HEX)		Begin blinking video field
A4483919 (HEX)		Begin secure video field
A4483D1D (HEX)		Left delimiter - right justify
A4483F1F (HEX)		Left delimiter
A4483C1C (HEX)		Left delimiter - protected
A4483E1E (HEX)		Right delimiter
A44D	Esc T	Roll Down
A44E	Esc S	Roll Up
A45257		Enter configuration mode
A454	Esc Z	Enable lowercase
A451	Esc X	Disable forms mode
A456	Esc >	Exchange cursor line with line above
A457	Esc W	Enable forms mode
A459	Esc Y	Disable lowercase
A43C c r	Esc " c r	Programmable cursor
A4 d d A4	Esc d d	The ASCII digits are converted to EBCDIC digits

Figure 6-2: Control A4 Codes

Default PF and PA Key Settings

The control codes described above that are assigned to the IBM 3270 PF and PA keys by default are as follows:

IBM 3270 Key (HEX)	Emulator Control Code	Function
PF1	A6	Spcfy key (transmits cursor position)
PF2	E8	Display status line
PF3	E7	Enter configuration mode
PF4	Ctrl space D Ctrl	Initialize terminal options
PF10	83	Insert line (using location of keyboard cursor)
PF11	85	Delete line (using location of keyboard cursor)
PA1		Displays next terminal page
PA2		Displays previous terminal page
PA3		Refreshes terminal page

Figure 6-3: IBM 3270 PF and PA Keys

Real-Time Configuration Mode

The PF3 key is assigned this function by default. When initiated the following screen is displayed:

```
                MT EMULATOR ConTRoL Mode Screen


Please Select one of the following options:

[ ] Video Options

[ ] Keyboard Options (not yet implemented)

[ ] Display Product Status

[ ] Reload Options from configuration file

[ ] Exit ConTRoL Mode
```

The Display Product Status option returns the version of the software with a list of the major features supported. The MT EMULATOR does not support all control sequences that might be found in the input data. If any unsupported sequences are found, they are listed on this screen to inform the user.

The Reload Options from configuration file causes the terminal options to be reloaded from the permanent options that were stored in the configuration file.

The Video Options Screen displays the current settings for that terminal. These options can be changed temporarily or permanently. For a detailed description of the options, see Chapter 2.

```
                    Video Option Screen

CHANGE TYPE: [ ] Temporary  Permanent   No change

EDIT:   [N] FULL FORMS PAGE XMIT      [N] LINE-AT-A-TIME XMI
        [N] LOWER CASE ENABLED        [N] DC1 CLEARS TO EOL
        [N] DCOM LF -> CR             [Y] DCOM CR -> LF
        [N] DC2 ADV DCP               [N] VT PAGE ADV
        [N] VAR TABS                  [N] TAB RETAIN
        [N] FF CLR TABS

MISC:   [Y] STATUS DISPLAY ENABLED    [N] FIELD FILL ENABLED
        [N] OPTION "I" ENABLED        [Y] LSCANDE ENABLED

        [00] CODE POINT CHART

        ALTERNATE DELIMITER (HEX):    [00] OPEN  [00] CLOSE
```

Make the modifications by entering either a Y (yes) or a N (no) against the listed options. Characters other than Y are interpreted to mean NO. In the alternate delimiter case, invalid hex characters will leave the settings unchanged, while "00" deletes any existing alternate delimiters. If the options are to be saved across MT EMULATOR termination, enter a *P* in the CHANGE TYPE field and press the Enter key. If the changes are to be temporary, enter a *T* in the CHANGE TYPE field and press the Enter key. Leaving this field blank or entering an *N* will leave the options unchanged.

Since the terminal options can be changed by programs, inquiring upon the configuration at random times may display option settings different from the initial configuration.

Customizing the PF Keys

Custom PF key definitions can be implemented by binding a procedure to the MT EMULATOR code file. A sample procedure is listed below. The code sequences in this file should be modified to reflect the needs of your site.

```
$ LEVEL 4
[EBCDIC ARRAY KEYSTROKEBUFFER[0]; ]
PROCEDURE BUILDPFKEYTABLES;
BEGIN
RESIZE(KEYSTROKEBUFFER, 256); % Resize depending on total %
                              % length of key program.    %
REPLACE KEYSTROKEBUFFER BY
    %
    %     length of key program including the length byte
    %
    %     followed by the keystroke codes and graphics.
    %
    %     Use ASCII codes for graphics
    %
    %     The keystroke buffer is loaded in ascending PF
    %
    %     key sequence. (PF 1 to PF 24)
    %
    %
    %
          48"02", 48"A6",              % SPCFY
          48"02", 48"E8",              % Status Line
          48"02", 48"E7",              % Config
          48"05", 48"A42044A4",        % CTRL sp D CTRL
          48"05", 48"A43030AC",        % CTRL 0 0 XMIT
          48"05", 48"A43039AC",        % CTRL 0 9 XMIT
          48"05", 48"A43838AC",        % CTRL 8 8 XMIT
          48"05", 48"A43738AC",        % CTRL 7 8 XMIT
          48"05", 48"A43930AC",        % CTRL 9 0 XMIT
          48"02", 48"83",              % Line Insert
          48"02", 48"85",              % Line Delete
          48"01",                      % Dead  Key
          48"05", 48"A43839AC",        % CTRL 8 9 XMIT
          48"05", 48"A43739AC",        % CTRL 7 9 XMIT
          48"05", 48"A43730AC",        % CTRL 7 0 XMIT
          48"05", 48"A43830AC",        % CTRL 8 0 XMIT
          48"05", 48"A43735AC",        % CTRL 7 5 XMIT
          48"05", 48"A43835AC",        % CTRL 8 5 XMIT
          48"05", 48"A43732AC",        % CTRL 7 2 XMIT
          48"05", 48"A43832AC",        % CTRL 8 2 XMIT
          48"05", 48"A43231AC"         % CTRL 2 1 XMIT
          48"05", 48"A43232AC",        % CTRL 2 2 XMIT
          48"05", 48"A43235AC",        % CTRL 2 5 XMIT
          48"05", 48"A43236AC"         % CTRL 2 6 XMIT
        ;

END BUILDPFKEYTABLES;
```

The following WFL job can be used to compile and bind this separately compiled procedure into the MT EMULATOR. The title of the disk file containing the procedure BUILDPFKEYTABLES is assumed to be BUILDPFKEYTABLES.

```
BEGIN JOB PFKEYS;
COMPILE OBJECT/BUILDPFKEYTABLES WITH ALGOL LIBRARY;
        COMPILER FILE CARD=BUILDPFKEYTABLES ON DISK;
BIND *ENVOY/OBJECT/LIBRARY/MTEMULATOR WITH BINDER LIBRARY;
?COMPILER DATA
HOST IS ENVOY/OBJECT/LIBRARY/MTEMULATOR;
BIND BUILDPFKEYTABLES FROM OBJECT/BUILDPFKEYTABLES;
STOP;
?END JOB
```

Chapter 7
MT EMULATOR Software Interface

The MT EMULATOR is an independent library module with three major entry points:

```
OPEN MTEMULATOR SESSION

CLOSE MTEMULATOR SESSION and

MTEMULATOR
```

OPEN MTEMULATOR SESSION

The ALGOL declaration for this procedure is:

```
REAL PROCEDURE OPENMTEMULATORSESSION
                (STATIONNAME,
                 STATIONNAMELENGTH);
     VALUE
                 STATIONNAMELENGTH;
     EBCDIC ARRAY
                 STATIONNAME[0];
     REAL
                 STATIONNAMELENGTH;
```

This Real procedure expects two parameters and returns a terminal index as procedure value. The parameters are the station name of the IBM 3270 device and the length of the station name.

Successful execution of the procedure results in procedure values greater than 0.

The MT EMULATOR tables are searched for by the station name. If a match is found, the associated terminal index is returned as procedure value. If no such station name exists in the tables, then the configuration file is searched for a match. If a record is found, the configuration is loaded for the station. If no configuration record exists for this station, one is created. In any event, the station name is added to the MT EMULATOR tables and a terminal index is returned.

After the successful execution of this procedure, the indicated station has a position in the MT EMULATOR tables for future dialogs.

The application calling this procedure is expected to save the value returned by this procedure, and use it in future communications with the MT EMULATOR regarding this station.

CLOSE MTEMULATOR SESSION

The ALGOL declaration for this procedure is:

```
BOOLEAN PROCEDURE CLOSEMTEMULATORSESSION
                  (STATIONNAME,
                   STATIONNAMELENGTH);
        VALUE
                   STATIONNAMELENGTH;
        EBCDIC ARRAY
                   STATIONNAME[0];
        REAL
                   STATIONNAMELENGTH;
```

This Boolean procedure expects two parameters and they are the same as the OPEN MTEMULATOR SESSION procedure.

The MT EMULATOR tables are searched for the station name and, if found, the station state information is forgotten and the record positions used by this station in the display store file are released to the available record pool.

MTEMULATOR

The ALGOL declaration for this procedure is:

```
INTEGER PROCEDURE MTEMULATOR
                  (EMULATORREQUEST,
                   TERMINALINDEX,
                   SOURCEBUFFER,
                   DESTBUFFER,
                   SOURCEOFFSET,
                   DESTOFFSET,
                   SOURCESIZE,
                   EMULATORRESULT);
        VALUE
                   EMULATORREQUEST,
                   TERMINALINDEX,
                   SOURCEOFFSET,
                   DESTOFFSET,
                   SOURCESIZE;
        EBCDIC ARRAY
                   SOURCEBUFFER,
                   DESTBUFFER [0];
        REAL
                   EMULATORREQUEST,
                   TERMINALINDEX,
                   SOURCEOFFSET,
                   DESTOFFSET,
                   SOURCESIZE,
                   EMULATORRESULT;
```

This Integer procedure is the main component of the MT EMULATOR library.

The calling application packs the information necessary for the MT EMULATOR in a request word and the MT EMULATOR returns the result via a result word. Unisys Corporation reserves the right to change the layout of these words at any time.

There are parameters for the source and destination of the message and their offsets within the arrays. There is another parameter indicating the length of the source message.

The length of the resulting message is the value of this procedure.

Request Word Layout

REQUESTBUGF = [47:01]

If this bit is turned on, then all procedural activities are written to a monitor file.

REQUESTAPPLYEDITORF = [43:01]

This bit, when set, invokes the logical io editor process before translating the incoming message.

REQUESTCARRIAGEF = [39:16]

This field is used when the REQUESTAPPLYEDITORF bit is set. It is a copy of the Carriage Control field employed in DCWRITES.

REQUESTTOGSF = [19:08]

This field is used when the REQUESTAPPLYEDITORF bit is set. It is a copy of the TOGs field found in DCWRITES.

REQUESTCANDEF = [03:01]

This bit, when set, invokes special output screen formatting.

REQUESTLOCALMODEF = [02:01]

This bit, when set, invokes input buffering. No output is generated but the input message is recorded in the keyboard screen area. When there is a transition out of this mode, the screen area is output as a result.

REQUESTDIALOGF = [01:01]

This bit must be set when calling the MTEMULATOR procedure if a previous call on the procedure for a message from a terminal resulted in an output to the terminal instead of to the application. (See Result Word Layout.)

REQUESTFROMTERMINALF = [00:01]

This bit is set if the message to be translated came from an IBM 3270. When this bit is off, the message is assumed to be from an application.

Result Word Layout

RESULTTOPROGRAMLENF = [19:16]

Some Unisys terminal orders require that output be sent to the terminal as well as the application. In those instances, data for the application follows the output to the terminal in the same output array. This field gives the length of the message to the application.

RESULTTOPROGRAMF = [01:01]

This bit is set by the MT EMULATOR, if the output is to go back to the application.

RESULTTOTERMINALF = [00:01]

This bit is set by the MT EMULATOR, if the output must go back to the terminal. If the input came from the terminal and the MT EMULATOR determined that the output must be returned to the terminal, then the MTEMULATOR procedure must be called one more time with REQUESTFROMTERMINALF and REQUESTAPPLYEDITORF reset and REQUESTDIALOGF set. Other options, like REQUESTCANDEF, are set as needed. This second call to the MT EMULATOR must not be made if RESULTTOPROGRAMF is also set.

Appendix A
Unisys Terminal Escape Sequences

The following sequences begin with the Esc code, hex 27 (the blanks are only for readability):

Esc Sequence	Description
Configuration Group	
R A hhhh kk	Alter scratchpad contents starting at address hhhh and ending at hhhh + kk - 1 with the data following kk (ASCII)
R H hhhh kk	Same as above, but the data following kk is hex pairs
R S kk <msg>	Alter application status line with <msg> for kk bytes
R C	Reconfigure scratchpad temporary
R P	Reconfigure scratchpad permanent
R T aaaa cc	Transmit scratchpad contents starting at address aaaa and ending at aaaa + cc - 1
SP D	Load terminal configuration from terminal configuration file
R B h k n	N/S
Cursor Control Group	
" c r	Move cursor to column c, row r (ASCII)
$ x	Move cursor to home position of page "x"
C	Move cursor right one position
&	Align keyboard cursor to data comm cursor
#	N/S
.	N/S

Esc Sequence	Description
Editing Group	
J	Clear to end of page
K	Clear to end of line
L	Line insert
M	Line delete
P	N/S
S	Roll page up
T	Roll page down
@	N/S
!	N/S
%	N/S
>	Exchange cursor line with line above
<	Exchange cursor line with line below
Forms Mode Group	
W	Enter forms mode request for data comm cursor page
X	Exit cursor page from forms mode
Key Programming Group	
R K	N/S
Printer Commands Group	
;	N/S
:	N/S
]	N/S

Esc Sequence	Description
Disk Commands Group	
A m	N/S
B m	N/S
G m	N/S
H m	N/S
I m	N/S
Q m	N/S
U nnn	N/S
V nnn	N/S
* m	N/S
+ m	N/S
, m	N/S
Search Mode Group	
E	N/S
F	N/S
- x	N/S
Terminal Control Group	
D	N/S
R L	N/S
SP V	Transmit MT EMULATOR version
Y	Lowercase disable
Z	Allow lowercase data entry
' x	N/S
_ x	N/S

Esc Sequence	Description
?	N/S
(	Transmit page to host

Character Set Group

SP C	Clear page

Video Control Group

O	N/S
N	N/S

Note: *SP means 'space' and N/S means 'not supported.'*

Appendix B
Unisys Terminal Configuration

Scratchpad Layout

Address	Bit	Function	Values
0080	7	Data comm character transmission mode	Always 0
	6	Data comm line control procedure	Always 0
	5	Action on reception of SOH	Always 0
	4	Transmission of forms mode page	1 - special forms xmit 0 - normal forms xmit
	3	DC1 interpretation	0 - clear from dcp to eol
	2	ODT flag	Always 0
	1	Spare	Always 0
	0	Parity check	Always 0
0084	0-7	Lines per page minus 1	Always 17
0085	0-7	Characters per line minus 1	Always 4F
0086	0-7	Alternate US character	Examples 5B - [7B - {
0087	0-7	Alternate RS character	Examples 5D -] 7D - }

Address	Bit	Function	Values
008D	0-7	Language	Always 1
0091	0-7	Total line count	48 - 72 lines
0092	7	Delay in dropping RTS	Always 0
	6	Lowercase lockout	1 - uppercase only 0 - upper and lowercase
	5	Cursor display	Always 0
	4	Auto-skip to next forms field	Always 0
	3	Number of lines on screen	Always 1
	2	Number of characters per line	Always 1
	1	Cursor mode	Always 0
	0	Data comm line feed interpretation	1 - move data comm cursor to col 1 next line 0 - move data comm cursor down one line
0093	0-7	Number of lines on screen	Always 17
0094 0095	0-15	Display store size in characters	Always 1680
009C 009D	0-15	Data comm buffer size decrement	Always 0000
009E 009F	0-15	Data comm buffer size in characters	Always 0780

Address	Bit	Function	Values
00A0	7	Xmit home	1 - current cursor line at column 1 0 - mobile home line
	6	RET Key: write CR into display store	Always 0
	5	RET Key: cursor motion	Always 0
	4	Data comm CR: write display store	Always 0
	3	Data comm CR: pointer motion	1 - to column 1 of same line 0 - to column 1 of next line
	2	Data comm ETX: pointer motion	Always 0
	1	Not Used	Always 0
	0	Data comm HT: write display store	Always 0

Notes

- All scratchpad locations and all data stored in locations 0080 thru 00A0 are expressed in hex notation.
- Writing into some of the locations produces no effect on the operation of the MT EMULATOR and reading these locations results in hex 00. Locations not mentioned below are such locations.
- If possible, the MT EMULATOR's scratchpad contents mirror the Unisys terminal's scratchpad contents. [1]

Index

A

Attention Identifier (AID) character, 31

B

buffer control orders, 32

C

CANDE forms, 41
CANDE page feature, 40
CLOSE MTEMULATOR SESSION
    procedure, ALGOL declaration, 56
COMS
    code file on release tape, 5
    module name, 5
configuring
    MT EMULATOR, 7
    Unisys terminals, Appendix B
Current Buffer Address (CBA) pointer, 32

D

data stream, editing by NDL II, 467
Display Store File
    assigning location, 8
    description, 6
    effect of location on memory
      requirements, 15
    effect of terminal pages on, 8
Do-Not-Enter (X) symbols and conditions,
    19-20

E

EBCDIC code point translation chart
    screen, 9
Erase Unprotected to Address order, 33
ERASE WRITE ALTERNATIVE command,
    28
ERASE WRITE command, 28

F

field
    attributes, 34
    types, comparison between Unisys
      terminals and IBM 3270s, 40
fields, protected/unprotected, 34
fixed output field
    IBM 3270, 40
    Unisys terminal, 40
forms mode
    fields, 40
    request (Escape-W control sequence),
      40

G

GATEWAY
    code file on release tape, 5
    module name, 5

I

IBM 3270
    data stream format, outbound and
      inbound, 27
    field attribute layout, 34
    control codes for emulating Unisys
      terminal functions, 47-50

inbound data stream Attention Identifier (AID), 31
invoking Unisys terminal functions, 47
order codes, description, 32
PF and PA key settings, 50
Write Control Character (WCC), 30
IBM 3270 commands
ERASE WRITE ALTERNATIVE, 28
ERASE WRITE, 28
READ BUFFER, 29
READ MODIFIED ALL, 29
READ MODIFIED BUFFER, 29
WRITE, 28
IBM 3270 display station
description, 17
handling CANDE pages, 41
keyboard functions, 21-25
operator messages, 18
video, 18
Insert Cursor order, 33
installation
MT EMULATOR, 5,
MT EMULATOR internal files, 6

K

keyboard functions, IBM 3270, 21-25
keys, IBM 3270 keyboard
backspace, 23
Clear, 21
cursor, 22
delete character, 22
Erase EOF, 21
ErInp, 21
insert mode, 22
newline, 24
program access, 23
program function, 23
Reset, 23
tab, 21, 22, 24
keys, PF, customizing, 53

L

left-justified field
IBM 3270, 40
Unisys terminal, 40

M

memory requirements, MT EMULATOR, 15
Modified Data Tag, 35
MT EMULATOR
application status line, 39
closing a session, 56
configuring, 7
control mode screen, 51
data comm cursor, 37
description, 3, 37
display buffer, 37
editor, comparison to NDL II, 46
installation, 5
internal files, installing, 6
keyboard cursor, 37
memory requirements, 15
non-forms page, 41
opening a session, 55
operational environment screen, 8
page formatting, 40
peripheral control commands, 44
request word layout, 57
result word layout, 58
search mode commands, 44
software interface, 55
system status line, 39
terminal configuration escape sequences, 45
terminal control functions, 44
terminal operations using data comm, 42
text editing functions, 42
translate table, 38
video character set, 38
video control escape sequences, 44
video highlights recognition, 38

MT EMULATOR procedure, ALGOL declaration, 56

N

NDL II editing of data stream, 46

O

OPEN MTEMULATOR SESSION
procedure, ALGOL declaration, 55
operational environment
screen, 8
setting, 8
outbound data stream
commands, 28
format, 27

P

page formatting, MT EMULATOR, forms mode, 40
PF key definitions, customizing, 53
print format orders, 32
Program Tab order, 33
protected transmittable field
IBM 3270, 40
Unisys terminal, 40

R

READ BUFFER command, 29
READ MODIFIED ALL command, 29
READ MODIFIED BUFFER command, 29
Readiness and system connection messages, 18
Repeat to Address order, 33
right-justified field, Unisys terminal, 40

S

screen images file (Display Store File), 6
screens
EBCDIC Code Point Translation Chart, 9
MT EMULATOR Configurator Home Menu, 7
MT EMULATOR Control Mode, 51
MT EMULATOR Station Options, 11
Operational Environment, 8
video option (for 3270 in emulation), 52
Set Buffer Address order, 32
SNA ENVOY
code file on release tape, 5
module name, 5
Start Field order, 32

T

Terminal Configuration File (TCF), 6
terminal options
changing or reading, 45
setting, 11–14
terminal pages, setting number
text editing, MT EMULATOR, 42–44
translate tables for local character codes, 9

U

Unisys terminals
escape sequences, Appendix A
field types, 40
scratchpad layout, Appendix B

W

work buffers, setting, 8
WRITE command, 28
Write Control Character (WCC), 30

X

X (Do-Not-Enter) symbols and conditions, 19–20
X?+ symbol, 20
XPROG4nn message, 19
XPROG7nn message, 19
XSYSTEM message, 20